Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application. No changes have been made herein.

Listing of Claims:

What is claimed is:

1. **(Previously Presented)** A method comprising:

providing error correction support for application data written in a markup description language that is to be exchanged between computing devices, said providing further comprising:

defining a tag set to prevent errors or incorrect character conversions that occur frequently during the exchange and re-input of text in a markup description language used to write data or sentences; and

using a tag set to add rewritten information to a predetermined portion of said application data written in a markup description language, in order that that the number of said errors or incorrect character conversions occurring during the re-input of text is reduced.

2. **(Original)** The error correction support method according to claim 1, wherein said tag set is defined for a character in the same shape, a similar character, a space or a character having a complicated shape.

3. **(Canceled)**

4. **(Previously Presented)** A method comprising:

providing error correction support for application data written in a markup description language that is to be exchanged between computing devices, said providing further comprising:

selecting, from elements of said application data written in a markup description language, a text portion that needs error correction support, said error correction related to errors comprising errors or incorrect character conversions that occur frequently during the exchange and re-input of text in a markup description language used to write data or sentences;

enclosing said selected text portion using predetermined tags; and

writing correction code, which is based on a predetermined algorithm, in said text portion enclosed by said predetermined tags,

in order that the number of said errors or incorrect character conversions is ultimately reduced.

5. **(Original)** The error correction support method according to claim 4, wherein said correction code is calculated for a character string that represents an attribute value or an attribute name, and is written using a predetermined attribute for the description of an error code.

6. **(Previously Presented)** A method comprising:

providing error correction support for application data written in a markup description language that is to be exchanged between computing devices, said providing further comprising:

selecting, from elements of said application data written in a markup description language, character strings that require error correction support;

generating, for said selected character strings, error correction codes that are based on a predetermined algorithm; and

writing said error correction codes as nodes for said application data written in descriptive markup language, in order that the number of selected character string errors is reduced.

7. **(Original)** The error correction support method according to claim 6, wherein said error correction codes are generated for all multiple character strings that are selected, and are added after predetermined elements of said application data have been written.

8. **(Previously Presented)** A method comprising:

providing error correction support for application data written in a markup description language that is to be exchanged between computing devices, said providing further comprising:

sorting, into predetermined attribute types, words in said application data written in a markup description language that may constitute barriers in a context process;

writing said attribute types to said application data using a predetermined tag set; and

transmitting or storing said application data with which said attribute types are included.

9. **(Original)** The error correction support method according to claim 8, wherein said words that are sorted into said predetermined attribute types and that may constitute barriers in said context process is at the least one of a set comprising proper nouns, alphabetic abbreviations, tag names, keywords that appear as element values, attribute names, keywords that appear as attribute values.

10. **(Previously Presented)** A computer system comprising:

a memory;

an arrangement which provides error correction support for application data written in a markup description language that is to be exchanged between computing devices, said arrangement further comprising:

a markup addition profile, which includes information used for replacing a predetermined portion of said application data with tags and/or information for calculating error detection/correction code for said predetermined portion;

a markup addition module, for replacing, by referring to said markup addition profile, said predetermined portion with tags, and/or calculating said error detection/correction code for said predetermined portion, and for adding to said application data, to generate application data using correction information, said tags and/or said error detection/correction code; and

an outputter which outputs said application data with correction information that is obtained by said markup addition module.

11. **(Original)** The computer system according to claim 10, wherein said markup addition profile includes information used to insert said error detection/correction code into said application data, or information used to add said error detection/correction code as an annotation at the end of said application data.

12. – 19. **(Withdrawn)**

20. **(Previously Presented)** A storage medium on which a computer stores a computer-readable program that permits said computer to perform:

providing error correction support for application data written in a markup description language that is to be exchanged between computing devices, said providing further comprising:

a process for defining a tag set to prevent errors and incorrect character conversions that tend to occur during the re-input of text that is included in application data written in a markup language; and

a process for adding, to a predetermined portion of application data, replacement information using said tag set and/or correction code based on a predetermined algorithm.

21. **(Previously Presented)** A storage medium on which a computer stores a computer-readable program that permits said computer to perform:

providing error correction support for application data written in a markup description language that is to be exchanged between computing devices, said providing further comprising:

a process for recognizing a tag set that encloses replacement information, and/or a correction code, to prevent errors or incorrect character conversions that tend to occur during the re-input of text information that is included in application data written in a markup language; and

a process for employing said tag set to replace predetermined text information in said application data.

22. **(Previously Presented)** A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform an

error correction support method for application data written in a markup description language, said method comprising the steps of:

providing error correction support for application data written in a markup description language that is to be exchanged between computing devices, said providing further comprising:

defining a tag set to prevent errors or incorrect character conversions that occur frequently during the re-input of text; and

using a tag set to add rewritten information to a predetermined portion of said application data.

23. **(Previously Presented)** The error correction support method according to claim 4, further comprising, at a time subsequent to the aforementioned steps, the steps of:

removing said correction code and said predetermined tags and returning said application data written in a markup description language to its original form.

24. **(Previously Presented)** The method according to claim 1 wherein said application data written in a markup description language comprises a character which has a shape so complicated that when produced by a low-resolution facsimile machine on paper, the image of the character appears solid and ambiguous; and further wherein said tag set is used to prevent errors commonly associated with re-input of said character.